Simple Data in Computers

Admin

- Anyone new?
- Lets check syllabus
- Be sure to get your tools set up
  - See resources page of website.
- Reading assignment: Read Chapter 2 of Python Crash Course for this slide set.

This slide set

- Today we will look at the simplest part of programming
  - For those of you who have done a little in another language will likely be very slow
  - But for those who have never seen programming, you will have to start learning a (very) different way of thinking
  - So ask lots of questions.

Reminder: the simplest python program

- Whatever is in print() gets printed to command line on screen
- If you want it printed as you see it in the program
  - Put what you want printed into quotes “”
  - Everything is quotes is a string
    - More later.

- `print(“Hello comp151”)`

Variables

- Many times we want to hold on to a value and use it later in the program
- We can store these into a variable.
- A variable is a named memory location that you can store a value in
  - Your book refers to them as labels that you can assign values to, be sure to include the part about assigning values to them.
- Create a variable in python by putting the name on the left of an equals sign and the value to put in the variable on the right

Variables in use I

- In this simple example, we first put the string hello comp151 into the variable message
- Then we print message
  - Note there is no “” around message
  - We want the value of the message variable to be printed.
- When executing python program
  - Start from top of file and go down line by line (for now)

- `message = “Hello Comp151”`
- `print(message)`

Variables in use II

- What happens when I run this program
  - Lets start our *“lucky volunteer”* program :-)

- `message = “hello comp151”`
- `print(message)`
- `message = “students!”`
- `print(message)`

Variables in use II

- What happens when I run this program
  - >hello comp151
  - >students!
- Lets see it in pycharm if we haven't done it already
- Then lets go over it on the board and see changes as we "hand execute"

- `message = "hello comp151"`
- `print(message)`
- `message = "students!"`
- `print(message)`

Getting Some input

- Interesting programs need user input
- Several ways to get user input
  - Like? (Lucky volunteer?)

Getting Some input

- Interesting programs need user input
- Several ways to get user input
  - Like? (Lucky volunteer?)
    - Type into text field on window
    - Mouse input
    - Touch input
    - Sensors
    - And old standby – type into command line
- Guess which is easiest
  - Though we will try some of the others by the end of the semester

Getting Some Input

- Get command line user input in python with input function
  - Input will print whatever is in the () to the screen
  - Then halt the program till the user presses enter/return
  - Then return whatever string the user typed before enter/return which you can store in a variable

- `yourname = input("what is your name")`
- `print("Hello", yourname)`
-
-
- `//note the there the "" are`

Let's try it

- Now you try a little more
  - Extend the last example to first ask for the users name, then ask for the user's major
  - Finally print the users name and tell them that comp151 will help with <print the major>

- References
  - **https://tartle.co/talk-python-to-me-how-python-can-change-the-world-special-guest-michael-kennedy/ (see starting at 5:27)**
  - **https://medium.com/progate/python-is-a-superpower-6d818777137c**

Naming Variable

- Rules for names
  - Variables (and most other ‘things’ in python) have to following these rules for names
    - Can use letters, numbers and the underscore _
      - Numbers can’t be first
    - Cannot use python “keywords”
    - Be careful about anything that starts with an underscore
- Rules of thumb:
  - Variable names should be lower case
  - And use underscore between words
  - Eg: `user_name = “jsantore”`

Strings

- Recall: Computers are kinda giant calculators
  - So they all work with?

Strings

- Recall: Computers are kinda giant calculators
  - So they all work with?
  - Yup, its numbers all the way down.
  - But we humans often want to work with letters/characters
  - In programming, a sequence or ‘string’ of letters/characters is in fact called a string.
  - Enclose the characters that you want to treat as a string in quotes

Quotes and Strings

- In python you can use either kind of quote
  - Unlike some languages
- Single quotes
- Double quotes
- Even three of each
- Lets go over when you would use each
  - In pycharm – take notes.

-
-
  _
- `‘this is a string’`
- `“this is also a string”`
- `‘’’this is a triple quoted string’’’`

Strings in variables

- When we type a string into a program it is called a literal string
- We can also put a string into a variable
  - Strings are objects,
    - Defer most discussion of objects
  - We care about objects because we can ask them to do ‘stuff’ for us
  - Take the string variable, then add the ‘.’ character, and the ‘function’ that we want to do something for us.
  - See next slide

Asking Strings to do something for you

- Let’s have the program ask the user for their favorite class and then “shout” it back to them, but moving it to all upper case

Asking Strings to do something for you

- Let’s have the program ask the user for their favorite class and then “shout” it back to them, but moving it to all upper case
  - First ask the user for the string

- `your_class = input(“what is your favorite class:”)`

Asking Strings to do something for you

- Let’s have the program ask the user for their favorite class and then “shout” it back to them, but moving it to all upper case
  - First ask the user for the string
  - Then ask the string to make an upper case copy

- `your_class = input(“what is your favorite class:”)`
- `loud_version = your_class.upper()`

Asking Strings to do something for you

- Let’s have the program ask the user for their favorite class and then “shout” it back to them, but moving it to all upper case
  - First ask the user for the string
  - Then ask the string to make an upper case copy
  - Now finally, lets print the output

- `your_class = input(“what is your favorite class:”)`
- `loud_version = your_class.upper()`
- `print(“YOUR FAVORITE CLASS IS”, loud_version)`

F-Strings

- Sometimes we would like to use a variable in the string,
  - Then we don’t have to do that weird string and variable thing in the print.
  - And we can put several variables into the f-string
  - To use an f-string put the ‘f’ character before the open quote of string
    - Then put the variables (or other python code) into curly braces { }
  - See example next slide

F-string example

- First lets ask about a class

```
class_name = input("what class are you taking next semester:")
```

F-string example

- First lets ask about a class
- Then lets ask for a number of credits
  - Notice that we have stored those into two different variables

```
class_name = input("what class are you taking next semester:")

num_credits = input("How many credits is it:")
```

F-string example

- First lets ask about a class
- Then lets ask for a number of credits
  - Notice that we have stored those into two different variables
- Finally we will print this out using an f-string
- Lets ask lots of questions
- Anything you want to try?

```
class_name = input("what class are you taking next semester:")

num_credits = input("How many credits is it:")

print(f"You are taking {class_name} next semester for {num_credits} credits")
```

Math in python

- Flashback to middle school math:
  - What is an integer?
  -

Math in python

- Flashback to middle school math:
  - What is an integer?
    - A whole number with no fractional or decimal part right?
  - What is a ‘real number’?

Math in python

- Flashback to middle school math:
  - What is an integer?
    - A whole number with no fractional or decimal part right?
  - What is a ‘real number’?
    - Any number with a decimal or not right? (but not imaginary)
  - And for pretty much anyone not in the sciences or math, you don’t really care about this distinction
    - Outside of science no one cares if you write 4.0 or just 4
- BUT COMPUTERS DO
  - Because the underlying representation for integers and ‘floating point numbers’ is very different.

Math in python

- Other than the integer/float distinction, math in in python mostly works as you would expect

```
# addition
4+3 # result is 7
# subtraction
4-3 # result is 1
# multiplication
4*3 # result is 12
# division
4/3 # result is 1.3333333 etc
```

Math in python

- Other than the integer/float distinction, math in in python mostly works as you would expect
- There are two unusual arithmetic operators though
  - // is “floor division” (it throws away the remainder
  - % is the “modulo operator”
    - Fancy way of saying it gets the remainder
    - Remember remainders from 3rd grade?

```
# addition
4+3 # result is 7
# subtraction
4-3 # result is 1
# multiplication
4*3 # result is 12
# division
4/3 # result is 1.3333333 etc
# floor division
4//3 # result is 1
# modul0
5%3 # results is 2 - the remainder of 5 divided by 3
```

Constants

- Some languages have a notion of a constant, which is like a variable which can never be changed
- Python doesn't have real constants.
- But programmers indicate they want something to be treated as a constant by using all CAPs for the name
  - NUMBER_OF_SENATORS = 100

Quality of life: Large numbers

- In relatively recent versions of python we can make easier to read numbers by using underscores in number literals (the ones you type into your program)
- big_number = 100000000
  - Quick, what is the value of big_number?

Quality of life: Large numbers

- In relatively recent versions of python we can make easier to read numbers by using underscores in number literals (the ones you type into your program)
- big_number = 10000000
  - Quick, what is the value of big_number?
- How about
  - big_number = 10_000_000
- Both work in python and for large numbers, the second is more readable

Comments

- In programming a comment is a bit of writing meant to be read by programmers rather than interpreted by the computer
- In beginning programming
  - Use comments to explain what you are trying to do
- For more advanced programmers
  - Use comments to explain why you did it this way and not another way.
- Comments in python
  - Use the # character
    - Anything after that character on the same line is part of the comment and ignored by python

Reading Assignment

- As a reminder Read chapter 2 in Python Crash Course